**---------------------------- Perceptron ----------------EXERCISE#1-----------------------------------------**

from sklearn import datasets

import numpy as np

iris=datasets.load\_iris()

X=iris.data[:,[2,3]] # only two features

y=iris.target

print(y)

print(X)

print(np.unique(y)) # unique value

###

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3,random\_state=0)

print(np.shape(X))

print(np.shape(X\_train))

print(np.shape(X\_test))

####

from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

sc.fit(X\_train)

X\_train\_std=sc.transform(X\_train)

X\_test\_std=sc.transform(X\_test)

####

from sklearn.linear\_model import Perceptron

ppn=Perceptron(eta0=0.1,random\_state=0)

ppn.fit(X\_train\_std,y\_train)# Fit model

y\_pred=ppn.predict(X\_test\_std)# check accurracy

print('misclassified samples: %d'%(y\_test!=y\_pred).sum())#c

from sklearn.metrics import accuracy\_score

print('Accuracy:%.2f'%accuracy\_score(y\_test,y\_pred))

print('Training Accuracy: ', ppn.score(X\_train\_std,y\_train))

print('Testing Accuracy: ', ppn.score(X\_test\_std,y\_test))

**----------------------------Home Assignment --------------------------------------------------------**

----------------------------Logistic Regression----------------EXERCISE#2-----------------------------------------

from sklearn import datasets

import numpy as np

iris=datasets.load\_iris()

X=iris.data[:,[2,3]] # only two features

y=iris.target

print(y)

print(X)

print(np.unique(y)) # unique value

###

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3,random\_state=0)

print(np.shape(X))

print(np.shape(X\_train))

print(np.shape(X\_test))

####

from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

sc.fit(X\_train)

X\_train\_std=sc.transform(X\_train)

X\_test\_std=sc.transform(X\_test)

####

from sklearn.linear\_model import LogisticRegression

lr = LogisticRegression(C=1000.0, random\_state=0)

lr.fit(X\_train\_std, y\_train)

y\_pred=lr.predict(X\_test\_std)

print('misclassified samples: %d'%(y\_test!=y\_pred).sum())#compute

from sklearn.metrics import accuracy\_score

print('Accuracy:%.2f'%accuracy\_score(y\_test,y\_pred))

#Pridict Class probability of 1st sample

x=lr.predict\_proba(X\_test\_std[0:1,:])

print("%f %f %f"%(x[0,0],x[0,1],x[0,2]))